DOCUMENT RESUME

ED 392 441 IR 017 732

AUTHOR	Barr, John
TITLE	Worlds: A Constructionist Project for Second Grade Students.
PUB DATE	95
NOTE	7p.; In: "Emerging Technologies, Lifelong Learning, NECC '95"; see IR 017 705.
PUB TYPE	Reports - Research/Technical (143) -- Speeches/Conference Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	Cognitive Style; Computer Uses in Education; *Constructivism (Learning); Elementary Education; *Experiential Learning; *Grade 2; Holistic Approach; Learner Controlled Instruction; *Programming; *Student Projects
IDENTIFIERS	*HyperCard; *HyperTalk

ABSTRACT

Constructionist learning is characterized by three principles: "learning-by-making," "holistic learning," and a freedom to choose learning styles. This paper describes a project with second grade homeschooling students that tested the feasibility of using HyperCard as a medium for constructionist learning. The "worlds" project took 6 weeks. The first 2 weeks were spent learning the basics of HyperCard/HyperTalk language and developing a simple prototype world. In the second phase of the project, students designed and created a world based on some subject they were studying. Two examples are described which demonstrate that students can successfully program in the HyperCard/HyperTalk environment to create stacks that mimic various worlds. The project demonstrated that constructionist ideas can be applied successfully in early education and that the HyperCard environment nourished student design styles. (Contains seven references.) (AEF)

Worlds: A Constructionist Project for Second Grade Students

by John Barr

Paper presented at the NECC '95, the Annual National Educational Computing Conference (16th, Baltimore, MD, June 17-19, 1995.

…, S. (1992). Putting case-based instruction into context: Examples from legal, business, and medical education. *Journal of the Learning Sciences* 2, 367–427.

paper

Worlds: A Constructionist Project for Second Grade Students

John Barr
Department of Mathematics and Computer Science
Ithaca College
Ithaca, N.Y. 14850
(607) 274-3579
barr@icsun.sunnet.ithaca.edu

Key words: constructionist, HyperCard, multimedia, elementary education

Abstract

This paper describes a project performed with second grade students that tested the feasibility of using HyperCard as a medium for constructionist learning. Students were introduced to HyperCard and allowed to use its features, including HyperTalk, a programming language, to develop imaginary worlds. Several of the resulting projects are described.

Background

The constructionist school of learning has sprung up from work done by Seymour Papert and his colleagues at the MIT

Media Lab [Harel & Papert, 1990]. It is characterized by three principle concepts, "learning-by-making", "holistic learning", and a freedom to choose learning styles. Several writers have suggested these characteristics make the constructionist approach especially appropriate for elementary aged children [Harel & Papert, 1990; Papert, 1993].

"Learning-by-making", for example, focuses students' learning on the creation of concrete objects. A student's project might involve writing logical programs that direct a computer to draw pictures, for example, but the outcome would be increased knowledge about math and logic. Intellectual development occurs as it becomes necessary for the advancement of the creation and the creative process provides the motivation and energy for learning. This is especially useful for children who tend to view their environment in very concrete ways [Papert 1980, Papert 1993].

"Holistic" learning also makes sense for children. In constructionism knowledge is not chopped into discrete subjects to be studied in 50 minute intervals, but is viewed as a vast sea of information and techniques that can be accessed as needed. A specific project provides a student with focus. Knowledge about all subject areas then evolves from the project. In a well designed project every branch of study, from math to art to social sciences, will be naturally investigated as that knowledge is needed to complete the project. This also corresponds to how children naturally view the world [Papert 1993].

Finally, learning is traditionally forced into specific learning styles. For example, students may be encouraged to decompose problems into smaller easier-to-solve pieces or to use formal abstractions to obtain solutions. Constructionism allows children learn in different manners and no particular approach may work for any one child. Thus a student may design her project in any manner, through top-down planning, through an improvisional manner that evolves from the tools the students have available (called *bricolage* [Papert 1993]), or from any style in between. Students can use formal abstraction techniques or can create their project from a series of prototypes. Any approach that works for the student is encouraged.

This paper describes an experiment that uses constructionist principles for the education of second grade homeschooling children. The goal was to determine whether these principles could successfully be applied to children of this age group and to investigate which principles seemed especially appropriate.

The rest of this paper will describe this experiment by first examining some previous experiments and drawing some guidelines from them. Next the Ithaca experiment will be described and some example projects will be examined. Finally, the paper will discuss some future directions and summarize the results.

Implementations

Constructionism has spawned several experiments, both large and small. Some of the more famous experiments include the Brookline Logo Project [Papert, et. al. 1979], Project Mindstorm and Project Headlight [Papert 1993 pp. 78 - 79 and pp. 50 - 51], and the Instructional Software Design Project (ISDP) (including Harel's Software Designer project, [Harel 1990]). These studies have focused on students of various skill levels but have, in general, concentrated on students in grades 4 and higher. Though the experiments vary greatly in their implementation, they share some characteristics that spring from the constructionist principles that underlie them.

The most obvious shared characteristic of the projects, for example, is they provide students with complete access to computers and make the computer project the center of curricular activities. Computer programs are never studied as separate subjects, but are instead used as tools for creating or exploring knowledge and art.

In addition to having complete access to computers, students are largely self directed. They work on projects independently, though they do exchange ideas with other students, and have long periods of undirected time which they can devote entirely to their project. Students determine the pace of learning and the particular direction that is taken to get to their goals. Indeed, even the goal itself is often self determined.

In summary, then, the principle characteristics of constructionist projects are:

1. Complete access to the necessary tools and resources, such as computers.
2. Projects as the principal source of curricular activity.
3. Self directed goals, designs and creations.
4. Long periods in which to develop creations.
5. Freedom to interact with other students.

The Ithaca Experiment

This section describes the *worlds* project, a constructionist experiment with Second grade homeschooling students. As detailed below, the experiment was based on constructionist principles and followed the characteristics of earlier projects. Students operated in a computer-rich environment, i.e., there was at least one computer per student.

The goal of the experiment was to use a card based development environment (*HyperCard* and the *HyperTalk* programming language) to allow students to develop real or imaginary worlds. In particular, each student was given the task of creating a "card world" that reflected a world they were interested in. For example, a student who liked knights and medieval Europe could choose to create a liege complete with a walled city and a castle. Students interested in space exploration could create our solar system with space shuttles and satellites. Those whose interests were closer to home could create a plan of their town with enlargements for each of the stores and offices in the town.

The project took a total of six weeks. The first two weeks were spent learning the basics of the HyperCard/HyperTalk language and developing a simple prototype world. Each student created a plan of their house in this phase, though the actual design of the house was left to the individual students. In general, the first card of the HyperCard stack was a floor plan of the house, with each succeeding card (or group of cards) detailing a specific room or section of the house. Buttons were placed on the floor plan and other cards to allow users to go from the floor plan to a particular room, from one room to

another, to create special effects (e.g. opening a refrigerator door), or to go from a room back to the floor plan. Students used small amounts of programming to relate buttons to these actions.

In the second phase of the project, students designed and created a world based on some subject that they were currently studying. There was no instruction during this phase, although there was always an instructor in the lab to provide support. In general, student stacks began with a card that gave an overview of their world. Successive cards expanded particular aspects of their world and provided more details. Examples of particular projects are provided in the next section.

These projects are primarily visual. Students create pictures of their world in varying detail and connect these pictures together through buttons. Students added writing and sound to their pictures, but essential features of the world were portrayed visually. Students at this age are use to communicating visually so pictures are a natural medium for them.

Students had a variety of tools available for their projects. The Hypertext programming language was used to connect cards, allow repetitive display of cards, and provide selective choice of cards. Clip Art and the drawing tools native to HyperCard were used to create pictures and background images. Sound tools allowed students to orally annotate their pictures. Finally, buttons and fields, endemic to HyperCard, were extensively used to connect cards and create special effects.

This project had several distinguishing characteristics. First, although it focused on a very distinct technology, HyperCard, the technology was flexible enough to support a variety of projects. Students could, and did, choose to create worlds from a number of different academic areas.

Second, the *worlds* project is a separate approach to learning and not just a technique to be used with traditional instructionist learning. Though *worlds* can be used to reinforce ideas learned in more traditional manners, it was conceived as a more comprehensive approach to learning. That is, students are engaged in *creating* and, as a result, learning is achieved. It is the world and the students' desire to expand that world that drives the learning of both programming and other academic fields.

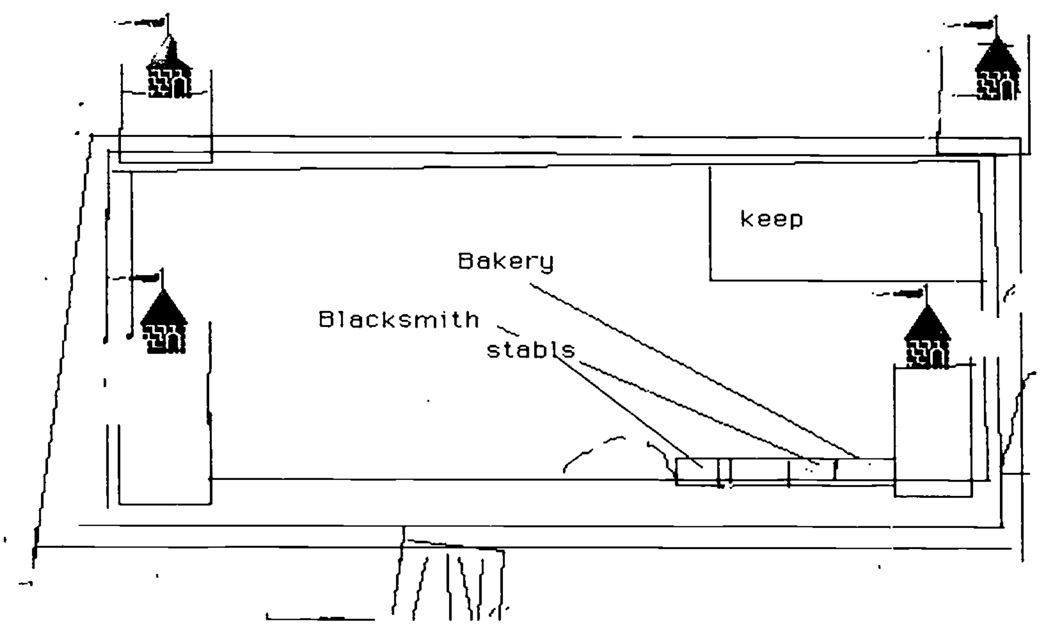
Third, the project was also constructionist in the design flexibility that it gave students. Students could, and did, approach their project from all ends of the design spectrum. Some students used a traditional top-down design, other students a more negotiational approach, and some used combinations.

Last, as in other constructionist projects, the goal of the project was not to provide specific knowledge to the students, but to let students learn what they needed to complete their project. Although there was initial instruction in the first two weeks of the project, the project was mostly self directed. Some of the knowledge learned was gained through experimentation, some through interaction with their peers, and some was provided, on request, by instructors.

Example Projects

Two examples highlight the success of the constructionist approach to this project, a stack created by a girl named Gwendolyn and one by a boy named Zachary. Gwendolyn's project was to illustrate medieval life by creating an example of a liege. The first card of the resulting stack is shown in figure 1. This card represents a typical city of the period. Each building on the card is actually a button that connects to a card (or cards) showing more detail.

Figure 1. First Card from Gwendolyn's Stack



Initially Gwendolyn's knowledge about medieval life drove the design. She took a top down design approach by first planning the layout of the liege and then making paper drawings of some of the cards. As the implementation developed, the design became less structured. Gwendolyn would create a card and then decide what should happen on the card. For example, she designed a card to represent stables and then decided what action individual horses would take when the user clicked on them.

Gwendolyn's implementations also drove her need for knowledge. For example, in the blacksmith shop she wanted a user to click on the tongs with the result that the tongs move to the table, pick up the horseshoe, take it to the furnace, and then place it back on the table. To create this effect, she had to learn how to make simple animations. The design drove her need for programming knowledge.

Similarly, when Gwendolyn wanted to create cards for the keep, she realized that she did not know what the inside of a keep looked like. She had to go to other resources (books on medieval life) to understand the design of the keep. Here her design drove her desire for knowledge about history.

Zachary approached his project, a water life world, with more structure. He initially designed his stack to have two sections, a fresh water life section and a sea life section. At every step of the implementation, he would complete the design of a section before he actually created that part of the stack.

From the beginning the project drove Zachary's interest in the related knowledge area, water life. After his initial design decision to create a partitioned stack, for example, Zachary realized that he didn't know which fish lived in fresh water and which lived in the sea.

Zachary also benefited from the free interaction in the class. When he saw other people using animation in their stacks, he immediately modified his design (again, he had to have a design before he began implementation) and set about learning the features of animation.

In general, Zachary was motivated by an interest in the computer. Although others focused on the project, learning to program only when it was needed in their design, Zachary was more driven to learn to program just to control the computer. This provided a nice balance in the classroom as Zachary was often the source of knowledge for programming techniques.

By the end of these project, both Zachary and Gwendolyn displayed a respectable grasp of a number of programming skills including simple conditional and repetition constructs. What is more interesting, however, is that both had also gained a fair amount of knowledge about different subjects (Medieval life and sea life) in the course of appropriating their programming skills.

Results

The project, though narrowly focused on homeschooled children, did successfully demonstrate that constructionist ideas can be applied even in early grades. Every student completed a stack reflecting her or his view of a world. Furthermore, learning was not restricted to a single field, but spanned mathematical/logical ideas (i.e. in the programming) as well historical, literate, and fantasy worlds. This knowledge was combined in a very natural way. Knowledge from each field was obtained as it was needed and from the most convenient source.

Though no formal testing mechanism was used either before or after the project, students did demonstrate greater mastery of programming and other fields at the completion of the project. Students were independently creating simple scripts in HyperTalk at the end of the project with a great fluency. They were also were able to orally articulate information about the academic area in which their world occurred in a sophisticated manner.

It is difficult to make generalizations about this project since the sample was so small. The project did demonstrate the feasibility of engaging younger students into computer centered projects and of integrating programming into other academic areas at this level. The key to this success is the flexibility of HyperCard. By providing a few preprogrammed buttons and access to clip art, even students that are unable to grasp sophisticated programming ideas can successfully create a world project. Thus students at all skill and ability levels will find success in the project.

In some sense, then, the *worlds* approach is better suited to younger students than other approaches such as logo or logo/Lego because of its emphasis on visual representation. The degree of logical sophistication to create successful worlds is quite low while visual representations can quickly and easily be created.

Future Directions

The most obvious extension to this experiment is to undertake larger and longer projects to test greater integration of *worlds* into a curriculum. The goal, of course, is to make a *worlds* project the centerpiece of a curriculum. Students would be challenged to create different worlds in areas where they have little of no knowledge. The creation of the worlds will drive the desire for more knowledge about the worlds.

Another direction for the *worlds* project would be to test it on older students. Some changes in emphasis might be made in this case, since *worlds* is currently very visually oriented to attract younger students. Older students might be encouraged to create, for example, a written description of their world on a separate stack that can run in parallel to their *worlds* stack.

Summary

This paper has described a project that was undertaken to determine whether younger students could use programming in a constructionist environment. The results seem to indicate that such an environment can be very successful, though larger

projects are necessary to test this hypothesis.

In particular, it was found that Second grade students can successfully program in the HyperCard/HyperTalk environment to create stacks that mimic various worlds. It was observed that students will independently seek to learn about programming and other fields when motivated by a creative task.

Finally, the design styles observed in other constructionist experiments are also possessed by students at this level. The HyperCard environment proved nourishing to all of these styles.

References

Harel, I. & S. Papert (1990). Software Design as a Learning Environment. *Interactive Learning Environments*, 1(1), pp. 1 - 25.

Kafai Yasmin, Idit Harel, Michael Resnick, Fred Martin, Elliot Soloway (1994). Constructionist Images of Programming, *Conference Proceedings, National Educational Computing Conference*, Boston, MA, 13 - 15 June, 1994. Pages 136 - 138.

Papert, S. (1993). *The Children's Machine*. New York: Basic Books.

Papert, S. (1991). Situating Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism*. Norwood, NJ: Ablex.

Papert, S. (1980). *Mindstorms*. New York: Basic Books.

Papert, S., D. Watt, A. diSessa, & S. Weir (1979). *Final report of the Brookline Logo projects. Part I and II.* (Logo memo No. 53). Cambridge MA: MIT.